Purpose: This course is designed as a continuation and enhancement of CPSC 124, Introduction to Programming. In that course, or its equivalent, you learned how to use various tools (Linux, javac, java, gedit, nedit, or perhaps an IDE such as Eclipse) to write, compile, run, and debug Java programs. You also began to learn about designing Java programs so they are modular, object-oriented, and well documented. I will insist that your programs for this class also have good design and implementation. Our goal is for you to begin to create more ambitious and fun programs while learning yet more features of the Java programming language.

Another equally important goal is beginning to develop a sense of quality in the programs you design and write. Of course we can get some tangle of code to compile and run but if it is buggy, does not scale, or has a poor interface, your great idea for a program will end up being just a bunch of bytes stored on a backup drive until it is deleted. A good attitude about programming is to design and write programs as if they are art works you want other people to enjoy and appreciate your skill.

Class Format: We will be looking at examples of Java programs in the MWF lecture sessions as we read through the various chapters in the textbook. I will be pointing out options we have for choices of data types, objects, methods, algorithms, and user interfaces to accomplish program goals. This will usually be in the context of a programming project you will be working on. Lab is a very good time to ask questions about Java and/or program design issues.

Class Behavior: I strongly discourage using laptops for note taking in class as the temptation to be distracted by email, Twitter, Reddit, Facebook, etc is usually too much to resist. I also insist you NOT spend your time checking your phone for texts, etc. You need to be paying attention and contributing to the ongoing class discussion.

Turn your phone off and put it away when you come into class.

Reading Materials: We will be reading most of the second half of David Eck's very nice *Introduction to Programming Using Java, Seventh Edition* textbook. This will fill in most of the remaining features forming the basis of the Java language. Java has evolved into a very versatile language offering a very complex API and there are a number of more advanced language features beyond this course which you may encounter in other computer science courses. Now is a good time to learn the Java toolset in some serious detail.

We will also be reading and working though the first few chapters of *Android 6 for Programmers* by Paul and Harvey Deitel. This book is chosen as a introduction to writing Android phone and tablet apps in Java. More on that below.

Materials: You will not need any specific devices for this course but you may wish to use your own laptop for programming assignments. If you have an Android smart phone or tablet, you may use that for the final project testing platform. You should have an Math/CS department Linux account. If not, you can get one by contacting Professor Eck.
Laboratory: Lab sessions are meant to provide time for hands-on experimentation and design of programming projects. Use the time to ask questions, try out new code, etc while there are people around to connect with. Again, you may bring your own laptop to lab if you wish and do your development on it but you are responsible for setting up your own Eclipse, NetBeans IDE or other development environment.

Projects: You will be designing, coding, and testing several fairly complex projects this term. Some will be individual projects, others in teams of two. Most of the projects are designed for running in a typical desktop or laptop system which has the Java Runtime Environment (JRE) installed. The last projects in this class will be different. You will be building apps for Android, the mobile OS project supported by Google. We will be using the IntelliJ Android Studio to develop mobile apps. If you have an Android phone or tablet, you should be able to run your app on that device. I will provide some Android tablets for teams that need them.

Collaboration: Lab assignments are your individual responsibility to complete using your own ideas and effort. You will certainly find yourself discussing labs, projects, etc with others in or out of the class but you MAY NOT SUBMIT ESSENTIALLY DUPLICATE WORK AND REPRESENT IT AS YOUR OWN WITHOUT ATTRIBUTION TO THE ORIGINATOR. Copying others intellectual work without attribution is a sign of deficiency on your part. It is more honest to submit your own, perhaps incomplete, code than borrow someone else’s work. Consequences for such behavior will include loss of credit for an assignment.

Quizzes: There is a quiz every Monday at the start of class. These are simply one or two questions on the material we have just covered in the previous week. Grades on quizzes are indicators of whether you are keeping up with and absorbing the content and tools in the course. The dozen or so quizzes each count 10 points toward your final grade so they are pretty much the equivalent of an hour exam.

I will not accept any late projects or quizzes, period. If you cannot attend class due to some genuine emergency then arrange to submit your (partially completed) project, hand it in to me before class, etc. Recognizing that emergencies and illness happen to us all, I will drop your lowest project and quiz grade when computing final grades.

Exams: There is only one exam in this course. The midterm exam is on Tuesday, October 17th during lab. There is no final exam but there will be presentations due in our final lab sessions on Tuesday December 12, 7 - 10pm. Be sure to check these dates BEFORE you make any travel plans.

Attendance: I will take attendance at every class and lab session. Missing more than three classes/labs will decrease your final course grade by at least one letter grade. The more absences, the more your grade decreases.

Grading: Your grade will be determined by all your activities in the class throughout the term. There is one midterm exam (100 points), 6-8 Java projects (160 points), about 12 in-class quizzes (120 points), one final Android project (60 points), and your attendance/participation. I will assign grades on the percentage of these points you amass during the term.

Disability Accommodations: If you are a student with a disability for which you may need accommodations, you should self-identify and register for services with the Coordinator of Disability Services at the Center for Teaching and Learning (CTL), and provide documentation of your disability. Disability related accommodations and services generally will not be provided until the registration and documentation process is complete. The guidelines for documenting disabilities can be found at the following website: http://www.hws.edu/disabilities

Please direct questions about this process or Disability Services at HWS to David Silver, Coordinator of Disability Services, at silver@hws.edu or 315-781-3351.